|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **SCHOOL OF COMPUTER SCIENCE AND ARTIFICIAL INTELLIGENCE** | | | | | **DEPARTMENT OF COMPUTER SCIENCE ENGINEERING** | | | | |
| **ProgramName:**B. Tech | | | | **Assignment Type: Lab** | | | **AcademicYear:**2025-2026 | | |
| **CourseCoordinatorName** | | | | Venkataramana Veeramsetty | | | | | |
| **Instructor(s)Name** | | | | 1. Dr. Mohammed Ali Shaik  2. Dr. T Sampath Kumar  3. Mr. S Naresh Kumar  4. Dr. V. Rajesh  5. Dr. Brij Kishore  6. Dr Pramoda Patro  7. Dr. Venkataramana  8. Dr. Ravi Chander  9. Dr. Jagjeeth Singh | | | | | |
| **CourseCode** | | | 24CS002PC215 | **CourseTitle** | | AI Assisted Coding | | | |
| **Year/Sem** | | | II/I | **Regulation** | | R24 | | | |
| **Date and Day**  **of Assignment** | | |  | **Time(s)** | |  | | | |
| **Duration** | | | 2 Hours | **Applicableto**  **Batches** | |  | | | |
| **AssignmentNumber:3.3**(Present assignment number)/**24**(Total number of assignments) | | | | | | | | | |
|  | | | | | | | | | |
|  | **Q.No.** | **Question** | | | | | | ***ExpectedTime***  ***to complete*** |  |
|  | 1 | Lab 3: Prompt Engineering – Improving Prompts and Context Management  **Lab Objectives:**   * To understand how prompt structure and wording influence AI-generated code. * To explore how context (like comments and function names) helps AI generate relevant output. * To evaluate the quality and accuracy of code based on prompt clarity. * To develop effective prompting strategies for AI-assisted programming.   **Lab Outcomes (LOs):**  After completing this lab, students will be able to:   * Generate Python code using Google Gemini in Google Colab. * Analyze the effectiveness of code explanations and suggestions by Gemini. * Set up and use Cursor AI for AI-powered coding assistance. * Evaluate and refactor code using Cursor AI features. * Compare AI tool behavior and code quality across different platforms.   **Task Description#1**   * Try 3 different prompts to generate a factorial function.   **Expected Output#1**   * Comparison of AI-generated code styles   PROMPT1:python code to generate the factorial functions.  INPUT:    OUTPUT:    **PROMPT 2 :develop a python program where the user is asked to enter a number and the program prints the factorial of the number**  **a python function which prints the factorial of 1 to 100**    **PROMPT 3:develop a python function which prints the factorial 1 to 100.**    **REPORT:**  **in the task1.py I have done it by just printing only factorial 5 but in task1\_2.py it asks the user to enter which number factorial we need after entering the number manually we get the result. Whereas in task1\_3.py it developed 1 to 10 factorials as per the prompt.**  **Task Description#2**   * Provide a clear example input-output prompt to generate a sorting function.   **Expected Output#2**   * Functional sorting code from AI   Prompt:  Give an example python code for sorting function.  Input:    **Output:**    **Task Description#3**   * Start with the vague prompt “Generate python code to calculate power bill” and improve it step-by-step   **Expected Output#3**   * Enhanced AI output with clearer prompts   PROMPT3:  **Write a python code to generate a power bill**  **INPUT:**    **OUTPUT:**    **PROMPT 2:**  ***# develop a python function which calculates the power bill for a given number of units and make it so that units cost different for different type of customers***  ***INPUT:***      **OUTPUT:**    **Report : in task3.py it asks enter the units and we enters it manually and then its gives the cost of current bill whereas in task3-1.py it asks us to enter the units after we need enter which type is it like residential , commercial or industrial then it gives the current bill as the output.**  **Task Description#4**   * Write structured comments to help AI generate two linked functions (e.g., login\_user() and register\_user()).   **Expected Output#4**   * Consistent functions with shared logic   PROMPT:    **OUTPUT:**    **Task Description#5**   * Analyzing Prompt Specificity: Improving Temperature Conversion Function with Clear Instructions   **Expected Output#5**   * Code quality difference analysis for various prompts * PROMPT 1:develop a python function for temperature conversion * INPUT,OUTPUT:   **PROMPT 2:**  ***#develop a python function for temperature conversion with clear instructions and print the result and add Reaumur scale***  **INPUT:**      **OUTPUT:**    **Report : in the task5.py I have done a temperature program in the basic way by adding 3 different scales like Celsius , Fahrenheit and kelvin scale it is basic kind but in the task5\_1.py I have added Reaumur scale too and in the output it asks the instruction like  1. Enter the temperature**  **2. from which scale**  **3. to which scale**  **If we add any thing invalid its shows invalid character**  **Note: Report should be submitted a word document for all tasks in a single document with prompts, comments & code explanation, and output and if required, screenshots**  **Evaluation Criteria:**   | **Criteria** | **Max Marks** | | --- | --- | | Factorial Function (Task#1) | 0.5 | | Sorting Function (Task#2) | 0.5 | | Vogue Vs. Specific Prompting (Task #3) | 0.5 | | Linked Functions (Task #4) | 0.5 | | Temperature Conversion Function (Task #5) | 0.5 | | **Total** | **2.5 Marks** | | | | | | | 03.08.2025 EOD |  |